



Detallando los datos de API

Transcripción

[00:00] Recibimos aquí en 405 método no permitido, pero en realidad esta URL también existe, vemos que es la misma que aquí que el delete, solamente que cambia el id. Y por ejemplo una cosa, que yo les comenté, es que la URL también depende del tipo de método http que yo use. Ahora ustedes podrán pensar: “A ver Diego, pero la URL existe, estamos llamándola aquí y no hemos implementado todavía este método con get”.

[00:34] ¿Entonces no debería dar un 404, not Found, que no existe? En este caso no, porque la URL sí existe. Por ejemplo, estoy usando una misma firma de URL, una misma firma el recurso, pero con un método diferente. Por lo tanto lo que acá me dice es que el método no está permitido porque mi back end no ha implementado ese método todavía. Entonces es por eso que yo recibo esta respuesta.

[01:02] ¿Qué es lo que toca hacer ahora? Implementarlo. Entonces venimos al código y vamos a crear un nuevo método. Para eso venimos aquí y como la firma y el método es muy parecida a la de mi método delete, yo voy a copiar este código, ya voy a borrar esto, no lo vamos a necesitar para este curso y lo voy a pegar. ¿Qué voy a hacer aquí?

[01:27] Primero que todo es un get, getMapping con id. Como es un get, no necesito transactional y voy a decir: retornaDatosMedico. ¿Por qué esto va a retornar los datos de un médico en específico? Nosotros ya estamos retornando los datos del médico en el listado, pero solo datos muy específicos, muy puntuales.

[01:52] Lo que yo quiero hacer ahora es retornar en sí todos los datos de un médico en particular. Para eso ya tengo mi `medicoRepository`, `getReferenceById`, no necesito hacer nada más, no lo voy a desactivar porque este método es un `get`, y lo que le voy a decir aquí es que devuelva un `ok`, pero si ven aquí, yo tengo dos tipos de `ok`.

[02:16] Tengo un `ok` que acepta un `body`, o sea que me acepta un parámetro y un `ok` que no me acepta nada. Entonces yo le puedo decir aquí que me resuelva un `ok` de médico y con esto sería suficiente. Pero nuevamente, yo no voy a retornar nunca objetos de mi entidad directamente. ¿Qué es lo que tengo que hacer aquí?

[02:38] Lo que puedo hacer aquí es simplemente decir `datosMedico` = aquí no es valor, aquí es `var`, `datosMedico`. Esto va a ser igual a `new datosRespuestaMedico` y como ya lo vimos acá, yo voy a copiar este código de aquí. Ya está. Y aquí en lugar de retornar médico, voy a retornar `datosMedico`.

[03:12] ¿Qué más voy a hacer? Aquí en lugar de `ResponseEntity` genérico le voy a decir que es un `ResponseEntity` de `datosRespuestaMedico`. Vengo aquí, guardo y guardo mi código. De esta forma vamos a ver que voy a borrar esta aquí porque no está compilando. Guardo acá.

[03:34] De esta forma ya tengo otro endpoint más para retornar los datos de un médico en específico. Vamos a ver si funciona. Entonces, voy a venir aquí. Voy a cambiarle el nombre aquí, voy a renombrado para obtener datos medico. Lo renombramos, perfecto, y le voy a dar un `send` y miren aquí.

[03:59] Ya me retornó los datos de Alexis Sandoval con código 200. ¿Esto qué quiere decir? Ahora tenemos un método más en la misma URL que ya es usada por el `delete`, pero con un verbo HTTP diferente. Y de esta forma, por ejemplo aquí en el `Post`, también, si guardamos otra vez otro dato en los headers que retorne ya va a estar disponible una vez que retorna este método, me retorne

location, ya está disponible este endpoint para poder acceder a este recurso en particular a través del método.

[04:34] Volvemos aquí. A nivel de código lo único que nos falta aquí en todo caso es este listado, porque no estamos haciéndolo con ResponseEntity, entonces aquí la cosa ya es mucho más fácil porque ya conocemos ResponseEntity. ¿Qué tengo que hacer?

[04:52] ResponseEntity de estoy de acá, vengo aquí. Todo bien. ¿Y aquí qué le digo? ResponseEntity.ok ¿y con cuál body? Es con este body que yo estoy retornando aquí, en este caso no hubo nada, nada, nada nuevo, no hubo nada del otro mundo. Guardamos. Dejamos que reinicie el servidor.

[05:22] Aquí fue lo mismo, solamente que lo que estamos haciendo ahora es ponerlo dentro de un wrapper. Así es como se llama eso, ResponseEntity. Es un wrapper, es un envoltorio, una envoltura. Vamos a darle aquí, listamos y tenemos el ok con los dos registros que tenemos ahora, solo que ahora está este retorno, que es dentro de un wrapper con envoltura para nuestra request HTTP.

[05:50] ¿Qué les pareció? Se ve el código un poco más entendible, se entiende lo que queremos retornar y ya estamos más acorde a las buenas prácticas de Rest. Como siempre, practiquen mucho, cualquier duda en el foro y nos vemos en la siguiente clase.